



แอปพลิเคชันพูดสื่อใจ

นางสาวนัฐทริกา ชีพันดุง

โครงงานนี้เป็นส่วนหนึ่งของการศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์

ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2559

ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

แอปพลิเคชันพุดสื่อใจ

นางสาวนัฐทริกา ชีพันดุง

โครงงานนี้เป็นส่วนหนึ่งของการศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์

ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2559

ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงการ	แอปพลิเคชันพุดสื่อใจ
โดย	นางสาวนัฐทริกา ชีพันดุง
อาจารย์ที่ปรึกษา	ดร.เกรียงศักดิ์ ตรีประพัฒน์
ระดับการศึกษา	วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์
ปีการศึกษา	2559

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์
คณะกรรมการสอบประเมินความรู้โครงการคอมพิวเตอร์

.....	อาจารย์ที่ปรึกษา
(ดร.เกรียงศักดิ์ ตรีประพัฒน์)	

.....	กรรมการ
(อาจารย์วาสนา เหง้าเกษ)	

.....	กรรมการ
(ดร.ทศพร จูฉิม)	

.....	หัวหน้าภาควิชา
(ดร.ทศพร จูฉิม)	

วันที่/...../.....

กิตติกรรมประกาศ

การพัฒนาแอปพลิเคชันชุดสื่อใจ บรรลุผลสำเร็จตามเป้าหมายได้ ด้วยความกรุณาและความช่วยเหลือจากหลายท่าน ข้าพเจ้าจึงใคร่ขอขอบพระคุณทุกท่าน ที่มีส่วนร่วมมา ณ โอกาสนี้

ขอขอบพระคุณ ดร.เกรียงศักดิ์ ตรีประพิณ อาจารย์ที่ปรึกษาโครงการ ผู้ที่คอยให้คำปรึกษาแนะแนวทางในการแก้ปัญหาต่าง ๆ และคอยให้กำลังใจ ตลอดการทำโครงการครั้งนี้

ขอขอบพระคุณ อาจารย์ทุกท่านในสาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ที่ให้คำแนะนำและถ่ายทอดความรู้ในการนำมาใช้ในการพัฒนาโครงการ และประกอบอาชีพต่อไปในอนาคต

ขอขอบคุณเจ้าหน้าที่ และบุคลากรของคณะวิทยาศาสตร์ ที่ได้อำนวยความสะดวกทางด้านอุปกรณ์ และเครื่องมือต่างๆ

ขอขอบพระคุณครอบครัวข้าพเจ้าที่ คอยให้กำลังใจและความห่วงใยเสมอมา ตลอดจนช่วยเหลือเรื่องทุนทรัพย์ทางการศึกษา

ขอขอบคุณเพื่อนสาขาวิทยาการคอมพิวเตอร์ รุ่นที่ 17 คณะวิทยาการคอมพิวเตอร์ มหาวิทยาลัยอุบลราชธานี ที่คอยให้คำแนะนำเกี่ยวกับโครงการ และคอยให้กำลังใจตลอดมา

กรกฎาคม 2560

นางสาวนัฐทริกา ชีพันธ์

โครงการงาน	แอปพลิเคชันพูดสื่อใจ
โดย	นางสาวนัฐทริกา ชีพพันธุ์
อาจารย์ที่ปรึกษา	ดร.เกรียงศักดิ์ ตรีประพัฒน์
ระดับการศึกษา	วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์
ปีการศึกษา	2559

บทคัดย่อ

ผู้แอปพลิเคชันพูดสื่อใจถูกพัฒนาให้ทำงานบน Android Operating System เน้นเป้าหมายการใช้งานสำหรับผู้พิการทางการได้ยิน เพื่อลดปัญหาการติดต่อสื่อสารในการดำเนินชีวิตประจำวัน จากปัญหาการรับรู้ทางระบบประสาทรับรู้เสียงของผู้พิการ ช่วยให้ผู้พิการทางการได้ยินสามารถสื่อสารกับผู้อื่นซึ่งไม่มีปัญหาทางการรับรู้เสียงได้ง่ายขึ้น เนื่องจากบุคคลทั่วไปที่ไม่มีปัญหาทางการรับรู้เสียงส่วนมากไม่มีความรู้เกี่ยวกับการสื่อสารด้วยภาษามือ ผู้พัฒนาจึงได้นำประโยชน์จากเทคโนโลยี Gesture มาใช้ในการพัฒนาสัญลักษณ์แทนข้อความเป็นส่วนช่วยในการเรียกใช้รายการข้อความจากฐานข้อมูล SQLite ของแอปพลิเคชัน ซึ่งสัญลักษณ์ทั้งหมดที่ใช้แทนรายการข้อความผู้ใช้สามารถกำหนดได้เอง ส่วนการแปลงข้อความเป็นเสียงสังเคราะห์ที่ผู้พัฒนาได้ใช้ Text to Speech API ของ Android มาช่วยในการพัฒนา และการแปลงเสียงภายนอกเป็นข้อความด้วย Speech To Text Engine เพื่อให้ผู้ร่วมสนทนากับผู้พิการสามารถสื่อสารกันเข้าใจกันได้ง่ายมากขึ้น

คำสำคัญ: Text to Speech, Gesture, Speech to Text, ผู้พิการทางการได้ยิน

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ผู้พิการทางการได้ยินแตกต่างจากบุคคลทั่วไปเพียงแค่การรับรู้เสียงทางระบบประสาทหูเท่านั้น ในเรื่องของการดำรงชีวิตหรือทำกิจกรรมทั่วไปสามารถทำร่วมกับบุคคลอื่นได้อย่างปกติ แต่เนื่องจากผู้พิการทางการได้ยินไม่สามารถรับรู้ทางเสียง ทำให้มีปัญหาในการติดต่อสื่อสารกับผู้อื่น แม้ปัจจุบันมีภาษามือเพื่อใช้ติดต่อสื่อสารกับผู้พิการทางการได้ยิน บุคคลทั่วไปที่มีประสาทการรับรู้ทางเสียงปกติส่วนมากไม่มีความรู้ในการสื่อสารด้วยภาษามือเพื่อใช้สื่อสารกับผู้พิการทางการได้ยิน ทำให้ผู้พัฒนาเล็งเห็นถึงปัญหานี้จึงได้คิดที่พัฒนาแอปพลิเคชันพูดสื่อใจขึ้นมา

แอปพลิเคชันพูดสื่อใจถูกพัฒนาเพื่อใช้บน Android Operating System เน้นเป้าหมายพัฒนาเพื่อผู้พิการทางการได้ยิน โดยผู้พัฒนาออกแบบให้แอปพลิเคชันมีความสามารถในการทำงาน 3 ส่วนหลัก คือ การแปลงข้อความเป็นเสียงในส่วนนี้แอปพลิเคชันทำงานจากการที่ผู้ใช้งานสัญลักษณ์บนหน้าจอ Smart Phone ซึ่งเป็นการใช้ความสามารถของเทคโนโลยี Gesture จากนั้นแอปพลิเคชันจะทำการดึงรายการข้อความที่ถูกแทนด้วยสัญลักษณ์นั้นมาแสดง เพื่อให้ผู้ใช้งานเลือกข้อความที่ต้องการ ต่อมาแอปพลิเคชันทำการแปลงข้อความเป็นเสียงสังเคราะห์ด้วย Text to Speech API ส่วนรายการข้อความที่ถูกแทนด้วยสัญลักษณ์ในแอปพลิเคชันผู้ใช้งานสามารถกำหนดได้เอง โดยแอปพลิเคชันมีส่วนจัดการกับสัญลักษณ์จากผู้ใช้งาน ซึ่งการข้อมูลจัดเก็บในฐานข้อมูล SQLite และการนำ GestureLibrary มาช่วยในการจัดเก็บ Gesture ในส่วนสุดท้ายการแปลงเสียงภายนอกเป็นข้อความเป็นการใช้ความสามารถของ Speech to Text Engine เพื่อความสะดวกในการสื่อสารกับบุคคลทั่วไป

1.2 วัตถุประสงค์

ช่วยอำนวยความสะดวกด้านการสื่อสารต่อผู้พิการทางการได้ยินกับผู้ร่วมสนทนาผ่าน Application บน Android Operating System

1.3 ขอบเขตของโครงการ

- 1) รองรับการทำงานบน Android Operating System version 4.2 ขึ้นไป

- 2) สำหรับผู้พิการทางการได้ยินใช้เป็นเครื่องมือช่วยสื่อสารกับผู้ร่วมสนทนา
- 3) สัญลักษณ์แทนข้อความและรายการข้อความในประโยคมาจากการกำหนดเองของผู้ใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ช่วยเป็นเสียงแทนผู้พิการทางการได้ยิน
- 2) ช่วยให้ผู้พิการทางการได้ยินได้รับสารเป็นข้อความตัวอักษร จากเสียงที่ผู้ร่วมสนทนาต้องการสื่อ

1.5 เครื่องมือที่ใช้ในการพัฒนา

1). ด้านฮาร์ดแวร์

- Lenovo G360
 - CPU : Intel Core i3 CPU M380 @2.53GHz x4
 - RAM : 4.00 GB
 - Hard Disk : 500 GB
 - Graphics : Intel Ironlake Moblie
 - Monitor : 14 inches
 - Operating System : Ubuntu 16.04
- Samsung Galaxy ACE 3
 - CPU : 1.2 GHz dual-core Qualcomm Krait
 - GPU : Qualcomm Adreno 305
 - RAM : 1.00 GB
 - Storage : 8 GB
 - Operating System : Android 4.2.2 Jelly Bean
- Lenovo Vibe C (a2020)
 - CPU : Quad-core 1.1 GHz Cortex-A7
 - GPU : Adreno 304
 - RAM : 1.00 GB
 - Storage : 8 GB
 - Operating System : Android 5.1.1 Lollipop

- Huawei Y6II
 - CPU : 1.2 GHz Krin620/MSM8952
 - GPU : Adreno 405,/ Mali 450 MP 4
 - RAM : 2.00 GB
 - Storage : 16 GB
 - Operating System : Android 6.0 Marshmallow

2). ด้านซอฟต์แวร์

- Android Studio เป็นเครื่องมือที่เอาไว้ใช้สำหรับพัฒนา Android Application

1.6 ขั้นตอนการดำเนินการ

อธิบายขั้นตอนการพัฒนาโครงการพร้อมทั้งแสดงตารางขั้นตอนการดำเนินโครงการ โดยให้ชื่อตารางแสดงอยู่ด้านบนของแต่ละตาราง และขีดทางด้านซ้ายของตาราง แสดงดังตารางที่ 1.1

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

แผนการดำเนินงานโครงการ	ธ.ค.59	ม.ค.60	ก.พ.60	มี.ค.60	เม.ย.60	พ.ค.60	มิ.ย. 60
1. ศึกษาความเป็นไปได้ของโครงการ	←→						
2. เสนอหัวข้อโครงการ		←→					
3. ศึกษาการใช้เครื่องมือในการพัฒนา		←→					
4. วิเคราะห์และออกแบบระบบ			←→				
5. พัฒนาระบบ			←→				
6. ทดสอบและแก้ปัญหา					←→		
7. จัดทำเอกสาร							←→

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีที่นำมาใช้ในการพัฒนาและตัวอย่าง Application ที่นำมาศึกษาเพื่อเป็นแนวทางในการพัฒนาแอปพลิเคชันชุดต่อไป รายละเอียดมีดังต่อไปนี้

2.1 Android System

2.2 ความรู้พื้นฐานเกี่ยวกับ Android API Packages หลักที่เกี่ยวข้อง

2.3 Gestures Builder

2.1 Android System

2.2.1 ความหมายของ Android Operating System

Android คือ ระบบปฏิบัติการแบบ Open Source โดย Google Inc. ในการเขียนโปรแกรมยึดหลักโครงสร้างภาษา Java และโปรแกรมที่พัฒนาจะทำงานอยู่ภายใต้ Dalvik Virtual Machine

Android Operating System เป็นระบบปฏิบัติการที่พัฒนาจากการนำเอาแกนกลางของ Linux Kernel Operating System ซึ่งเป็นระบบปฏิบัติการที่ออกแบบมาเพื่อทำงานเป็นเครื่อง Server มาพัฒนาต่อ เพื่อให้กลายเป็นระบบปฏิบัติการบน Mobile Operating System

2.2.2 ประเภทของ Android Operating System

เนื่องจาก Android Operating System เป็น Software ระบบเปิด จึงอนุญาตให้นักพัฒนาหรือผู้ที่สนใจสามารถ Download Source Code ได้ จึงทำให้ผู้พัฒนาหลายฝ่ายนำ Source Code มาปรับแต่งและพัฒนาสร้าง Application บน Android Operating System ของตัวเองมากขึ้นโดยสามารถแบ่ง Android Operating System ออกเป็น 3 กลุ่มประเภทใหญ่ ดังต่อไปนี้

- 1) Android Opensource Project (AOSP) เป็นระบบปฏิบัติการแรกที่ Google เปิดให้สามารถ Download Source Code ไปติดตั้งและใช้งานในอุปกรณ์ต่างๆ ได้โดยไม่เสียค่าใช้จ่าย
- 2) Open Handset Mobile (OHM) เป็น Android ที่ได้รับการพัฒนาร่วมกับ Open Handset Alliances (OHA) ซึ่งบริษัทเหล่านี้จะพัฒนาระบบ Android

ในแบบฉบับของตนเอง โดยมีรูปร่างหน้าตาการแสดงผลที่แตกต่างกันรวมไปถึง อาจจะมีเอกลักษณ์และรูปแบบการใช้งานเป็นของแต่ละบริษัท และ Android Program ประเภทนี้ก็จะได้รับสิทธิบริการเสริมจาก Google ที่เรียกว่า GMS (Google Mobile Service) ซึ่งเป็นบริการเสริมที่ทำให้ Android มีประสิทธิภาพมากขึ้น

- 3) Cooking or Customize เป็นระบบ Android ที่นักพัฒนานำเอา Source Code จากแหล่งต่างๆมาปรับแต่งให้อยู่ในแบบฉบับของตนเอง ซึ่งการพัฒนาต้องปลดล๊อคสิทธิในการใช้งานอุปกรณ์ เสียก่อนจึงจะสามารถติดตั้งได้ ทั้งนี้ระบบ Android ประเภทนี้ถือเป็นประเภทที่มีความสามารถสูงสุด เนื่องจากได้รับการปรับแต่งขีดความสามารถให้มีความเข้ากันได้กับอุปกรณ์จากผู้ใช้งานจริง

2.2.3 สถาปัตยกรรม Android

โครงสร้างสถาปัตยกรรม Android ตามมาตรฐานในการจัดแบ่งระบบ แบ่งออกเป็น 5 ส่วน ดังแสดงในภาพที่ 2.1 โดยมีรายละเอียดดังนี้



ภาพที่ 2.1 โครงสร้างสถาปัตยกรรม Android

(ที่มา : <http://www.sourcecode.in.th/articles.php?id=71>)

1) แอปพลิเคชัน (Application)

ส่วน Application หรือส่วนของโปรแกรมที่มีมากับระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ดโปรแกรมเอาไว้

2) แอปพลิเคชันเฟรมเวิร์ค (Application Framework)

เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยาก เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งานแอปพลิเคชันเฟรมเวิร์คในส่วนที่ต้องการใช้งาน แล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน ตัวอย่างเช่น

- Activities Manager เป็นกลุ่มของชุดคำสั่งที่จัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม (Activity)
- Content Providers เป็นกลุ่มของชุดคำสั่ง ที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่น และสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้
- View System เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับการจัดการโครงสร้างของหน้าจอที่แสดงผลในส่วนที่ติดต่อกับผู้ใช้งาน (User Interface)
- Telephony Manager เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลด้านโทรศัพท์ เช่นหมายเลขโทรศัพท์ เป็นต้น
- Resource Manager เป็นกลุ่มของชุดคำสั่งในการเข้าถึงข้อมูลที่เป็น ข้อความ, รูปภาพ
- Location Manager เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับตำแหน่งทางภูมิศาสตร์ ที่ระบบปฏิบัติการได้รับค่าจากอุปกรณ์
- Notification Manager เป็นกลุ่มของชุดคำสั่งที่จะถูกเรียกใช้เมื่อโปรแกรม ต้องการแสดงผลให้กับผู้ใช้งาน ผ่านทางแถบสถานะ(Status Bar) ของหน้าจอ

3) ไบเบอรี (Libraries)

เป็นส่วนของคุณค่าที่พัฒนาด้วย C/C++ โดยแบ่งคุณค่าออกเป็นกลุ่มตามวัตถุประสงค์ของการทำงาน เช่น Surface Manage จัดการเกี่ยวกับการแสดงผล, Media Framework จัดการเกี่ยวกับการการแสดงผลภาพและเสียง, Open GL | ES และ SGL จัดการเกี่ยวกับภาพ 3 มิติ และ 2 มิติ, SQLite จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น

4) Android Runtime

Android Runtime จะมี Dalvik Virtual Machine ที่ถูกออกแบบมาเพื่อให้ทำงานบนอุปกรณ์ที่มีหน่วยความจำ (Memory), หน่วยประมวลผลกลาง (CPU) และพลังงาน (Battery) ที่จำกัด ซึ่งการทำงานของ Dalvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงาน ไปเป็นไฟล์ .DEX ก่อนการทำงาน เหตุผลก็เพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับ หน่วยประมวลผลกลางที่มีความเร็วไม่มาก ส่วนต่อมาเป็น Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและคำสั่งสำคัญ โดยถูกเขียนด้วยภาษาจาวา (Java Language)

5) Linux Kernel

เป็นส่วนที่ทำหน้าที่หัวใจสำคัญ ในจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดย Android ได้นำเอาส่วนนี้มาจาก Linux 2.6 Kernel

2.2.4 องค์ประกอบของ Android แอปพลิเคชัน

องค์ประกอบของ Android แอปพลิเคชัน แบ่งออกเป็น 4 ประเภทใหญ่ ได้แก่ Activity, Service, Content Provider และ Broadcast Receiver ซึ่งองค์ประกอบของ Android มีเป้าหมาย การกระตุ้นการทำงาน และวงจรชีวิตต่างกัน โดยมีรายละเอียดในการใช้งาน แบ่งออกเป็น 4 ส่วน ดังต่อไปนี้

- 1) Activity คือ Application Component ที่ใช้ในการควบคุมการสร้าง User Interface เช่น การแสดงผลหน้าจอรายการอีเมล, การแสดงผลหน้าจอแบบฟอร์ม การส่งอีเมล เป็นต้น รวมถึงควบคุมการมีปฏิสัมพันธ์ระหว่างผู้ใช้กับ User Interface ด้วย เช่น เมื่อผู้ใช้เลือกรายการอีเมลก็จะทำการตอบสนองผู้ใช้โดย

การแสดงผลรายการอีเมลที่เลือก เป็นต้น สำหรับการสร้าง Activity นั้นทำได้โดยการสร้าง Class และให้สืบทอดจาก Class Activity หรือสืบทอดจาก Class ใด ๆ ก็ตามที่ได้รับสืบทอดมาจาก Class Activity โดย Activity หนึ่ง ๆ จะควบคุมการแสดงผล User Interface หนึ่ง ๆ เท่านั้น และนั่นแสดงให้เห็นว่า Application หนึ่ง ๆ จะประกอบด้วย Activity จำนวนมากที่ทำงานร่วมกันอยู่ อย่างไรก็ตามถึงแม้ว่า Activity จะทำงานร่วมกัน แต่ Activity เหล่านั้นยังคงเป็นอิสระจากกัน

- 2) Service คือ Application Component ที่ไม่มี User Interface และจะทำการประมวลผลใน Background กล่าวคือเป็นการประมวลผลที่ดำเนินไปพร้อมกับที่ผู้ใช้สามารถไปใช้งาน Application อื่น ๆ ได้ หรือกล่าวอีกมุมหนึ่ง การประมวลผลใน Background คือการประมวลผลที่สามารถทำงานขนานกันกับการทำงานอื่น ๆ ของผู้ใช้ ทั้งนี้ก็เพื่อทำให้เกิดการทำงานใด ๆ โดยที่ผู้ใช้ไม่จำเป็นต้องอยู่ในหน้าจอ นั้น ๆ ได้ ซึ่งอาจเป็นเพราะการทำงานนั้นต้องใช้ระยะเวลา เช่น การใช้ Service เปิดเพลง เพื่อให้ผู้ใช้สามารถไปใช้ Application อื่น ๆ ได้ แต่เพลงยังคงเล่นอยู่ หรือ การใช้ Service ดาวน์โหลดข้อมูลใด ๆ ที่มีขนาดใหญ่ เพื่อให้ผู้ใช้สามารถไปใช้ Application อื่น ๆ ได้ แต่การดาวน์โหลดยังคงดำเนินอยู่ เป็นต้น สำหรับการสร้าง Service นั้นทำได้โดยการสร้าง Class และให้สืบทอดจาก Class Service หรือสืบทอดจาก Class ใด ๆ ก็ตามที่ได้รับสืบทอดมาจาก Class Service
- 3) Content Provider คือ Application Component ที่ทำหน้าที่ในการควบคุมข้อมูลใด ๆ ของ Application ที่ต้องการ Share ให้ Application อื่น ๆ สามารถนำข้อมูลนั้น ๆ ไปใช้งานได้ หรือกล่าวในทางกลับกันก็คือ Application ใด ๆ สามารถนำข้อมูล (รวมถึงแก้ไขข้อมูลได้ ถ้า Content Provider อนุญาต) ของ Application อื่น ๆ มาใช้งานได้ โดยกระทำผ่าน Content Provider เช่น System ได้จัดเตรียม Content Provider ที่เป็นข้อมูลรายชื่อผู้ติดต่อ (Contact) ไว้ เพื่อให้ Application ที่ต้องการใช้ข้อมูลรายชื่อผู้ติดต่อนี้สามารถนำข้อมูลไปใช้หรือแก้ไขข้อมูลได้ เป็นต้น สำหรับการสร้าง Content Provider นั้น ทำได้โดยการสร้าง Class และให้สืบทอดจาก Class

ContentProvider หรือสืบทอดจาก Class ใด ๆ ก็ตามที่ได้รับการสืบทอดมาจาก Class ContentProvider

- 4) Broadcast Receiver คือ Application Component ที่ไม่มี User Interface โดยจะทำหน้าที่รับรู้สิ่งที่เกิดขึ้นของ System และนำมาบอกให้ผู้ใช้ได้รับรู้ เช่น เมื่อ Battery ต่ำ, เมื่อหน้าจอถูก Capture, เมื่อมีการพักหน้าจอ เป็นต้น ทั้งนี้ Application ใด ๆ สามารถนำ Broadcast Receiver มาใช้ประโยชน์ได้ เช่น เมื่อ Application ได้ Download ข้อมูลเสร็จเรียบร้อยแล้ว เป็นต้น ซึ่งโดยส่วนมากแล้วการตอบสนองของ Broadcast Receiver จะกระทำผ่าน Notification เพื่อแจ้งสิ่งที่เกิดขึ้นให้ผู้ใช้ได้รับรู้ สำหรับการสร้าง Broadcast Receiver นั้น ทำได้โดยการสร้าง Class และให้สืบทอดจาก Class BroadcastReceiver หรือสืบทอดจาก Class ใด ๆ ก็ตามที่ได้รับการสืบทอดมาจาก Class BroadcastReceiver

2.2 ความรู้พื้นฐานเกี่ยวกับ Android API Packages หลักที่เกี่ยวข้อง

2.2.1 Speech Recognition

Speech Recognition หรือ Voice Recognition เป็นการแปลงเสียงเป็นข้อความ อธิบายการใช้งานได้ดังต่อไปนี้

- 1) การเรียกใช้งาน Speech Recognition

การเรียกใช้งาน Speech Recognition สามารถเรียกใช้งานด้วย Intent และคำสั่ง StartActivityForResult ดังภาพที่ 2.2

```
1 import android.app.Activity;
2 import android.content.Intent;
3 import android.os.Bundle;
4 import android.speech.RecognizerIntent;
5 public class MainActivity extends Activity implements View.OnClickListener{
```

ภาพที่ 2.2 การเรียกใช้งาน Speech Recognition

```

8      private final static int REQUEST_VOICE_RECOGNITION = 10001;
9      ....
10     private void callVoiceRecognition(){
11         Intent intent = new
            Intent(Recognizer.ACTION_RECOGNIZE_SPEECH);
12         startActivityForResult(intent, REQUEST_VOIC_RECOGINITION);
13     }
14     @Override
15     public void onActivityResult(int requestCode, int resultCode, Intent
        data){
16         super.onActivityResult(requestCode, resultCode, data);
17
18         //ผลลัพธ์ที่ส่งกลับมา
19     }
20 }

```

ภาพที่ 2.2 การเรียกใช้งาน Speech Recognition (ต่อ)

ในบรรทัดที่ 6 กำหนด REQUEST_VOICE_RECOGNITION เป็น 10001 เพื่อใช้ตรวจสอบการส่งไปและส่งกลับมาของข้อมูล

2) การแปลงข้อความจากเสียง

การดึงข้อมูลผลลัพธ์ สามารถดึงออกมาเป็นข้อความในรูปแบบ ArrayList ได้
ดังภาพที่ 2.3

```

1  protected void onActivityResult(int requestCode, int resultCode, Intent data){
2      super.onActivityResult(requestCode, resultCode, data);
3      if(requestCode == REQUEST_VOIC_RECOGINITION &&
4          resultCode == RESULT_OK && data != null){

```

ภาพที่ 2.3 การแปลงข้อความจากเสียง

```

5          ArrayList<Strign> resultList == data.getStringArrayListExtra(
6              RecognizerIntent.EXTRA_RESULTS
7          );
8      }
9  }

```

ภาพที่ 2.3 การแปลงข้อความจากเสียง (ต่อ)

3) การกำหนดภาษาสำหรับการรับค่า

การกำหนดภาษาสำหรับการรับข้อความเสียงจากภายนอก สามารถกำหนดค่าให้ตัวแปร Intent ดังภาพที่ 2.4

```

1  Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
2  intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, "th-TH");
3  startActivityForResult(intent, REQUEST_VOIC_RECOGINITION);

```

ภาพที่ 2.4 การกำหนดภาษาสำหรับการรับค่า

2.2.2 Text To Speech

Text To Speech เป็นการแปลงข้อความออกเป็นเสียงสังเคราะห์ ผ่าน Text to Speech Engine

1) การเรียกใช้งาน Text to Speech

การเรียกใช้งาน Text to Speech สามารถทำได้โดยการเรียกใช้งาน class Text to Speech โดยการ import android.speech.tts.TextToSpeech; และทำการ Initialize ได้ดังภาพที่ 2.5

```

1  import android.app.Activity;
2  import android.os.Bundle;
3  import android.speech.tts.TextToSpeech;

```

ภาพที่ 2.5 การเรียกใช้งาน Text to Speech


```

4  public class MainActivity extends Activity implements
    TextToSpeech.OnInitListener{
5      private TextToSpeech tts;
6      @Override
7      protected void onCreate(Bundle savedInstanceState){
8          super.onCreate(savedInstanceState);
9          setContentView(R.layout.activity_main);
10
11         tts = new TextToSpeech(this, this);
12     }
13     @Override
14     public void onInit(int status){
15         //Do something here
16     }
17     @Override
18     protected void onDestroy(){
19         super.onDestroy();
20         tts.shutdown();
21     }
22 }

```

ภาพที่ 2.5 การเรียกใช้งาน Text to Speech (ต่อ)

2) การระบุ Text to Speech Engine

การระบุ Text to Speech Engine ที่ต้องการนำมาใช้งานในระบบปฏิบัติการ Android จำเป็นต้องมี Engine นั้น ยกตัวอย่างเช่น ต้องการระบุเป็น Google Text to Speech Engine ได้ดังนี้

```
tts = new TextToSpeech(this, this, "com.google.android.tts");
```

3) การตรวจสอบการ Initialize

เมื่อ Initialize ถูกเรียกใช้งานแล้วต้องการตรวจสอบการ Initialize ผลลัพธ์ที่เป็นไปได้มีอยู่ 2 กรณี คือ TextToSpeech.SUCCESS และ TextToSpeech.ERROR ตัวอย่างการตรวจสอบสามารถเขียนได้ดังภาพที่ 2.6

```
1 @Override
2 public void onInit(int status){
3     if(status == TextToSpeech.SUCCESS){
4         // Do something here
5     }
6 }
```

ภาพที่ 2.6 การตรวจสอบการ Initialize

4) การกำหนดภาษา

การกำหนดภาษาที่ต้องการรับค่าทำได้โดยใช้ setLanguage เช่น ตัวอย่าง ดังนี้

```
tts.setLanguage(Locale.ITALY);
```

5) การเรียกใช้งาน

เมื่อต้องการเรียกใช้งานให้แปลงข้อความเป็นเสียงสังเคราะห์ สามารถใช้คำสั่ง ดังนี้

```
tts.speak(String message, int queueMode, HashMap<String, String> params);
```

โดยที่

queueMode คือ การกำหนดวิธีการทำงานของ Text To Speech สามารถ

กำหนดได้ 2 แบบ คือ

- TextToSpeech.QUEUE_FLUSH : ถ้ามีคำสั่ง Speak ทำงานอยู่ก่อนหน้า จะหยุดทำงานทันที แล้วทำงานคำสั่ง Speak ตัวใหม่
- TextToSpeech.QUEUE_ADD : ถ้ามีคำสั่ง Speak ทำงานอยู่ก่อนหน้า จะรอคำสั่งก่อนหน้าทำงานจนเสร็จก่อนแล้วจึงทำงานต่อ

- params : ส่วนที่มีสำหรับส่งค่าให้กับ Text to Speech ซึ่งถ้าไม่มีให้
กำหนดค่าเป็น Null

ตัวอย่างการเรียกใช้งาน ดังนี้

```
tts.speak("ข้อความที่ต้องการ", TextToSpeech.QUEUE_FLUSH, null);
```

2.2.3 Gesture

Gesture ใน Android มีความหมายว่า ผู้ใช้สามารถใช้นิ้ววาดสัญลักษณ์ลงบนพื้นที่ที่กำหนดไว้ในแอปพลิเคชัน เพื่อให้แอปพลิเคชันทำตามคำสั่งที่ตั้งไว้

Gestures รองรับการพัฒนาโปรแกรมอยู่ 2 ตัว คือ

- 1) GestureDetector เป็นการสร้าง Event ที่ใช้ method ต่างๆ เข้ามาตรวจสอบ เช่น onFling, onDown, onLongPress, onScroll, onShowPress และ onSingleTapUp โดย method ที่กล่าวมาจะ implement มาอัตโนมัติหลังจาก implements OnGestureListener
- 2) GestureOverlayView เป็นการออกแบบเหตุการณ์ที่เกิดขึ้นหน้าจอในรูปแบบต่างๆ เช่นการ Slide ไปยังหน้าจอ หรือการวาดสัญลักษณ์บนหน้าจอ โดยระบุเงื่อนไขของการกระทำ

2.2.4 SQLite Database

SQLite Database คือฐานข้อมูลขนาดเล็ก รูปแบบการทำงานของ SQLite เป็นแบบ Standalone ทำงานอยู่ใน Application นั้น ซึ่งใน Android มี Package คือ android.database ซึ่งเป็น Library ที่สามารถเรียกใช้งานเกี่ยวกับ Database ได้

ในการนำคำสั่ง SQLite มาใช้งานในการสร้าง Android แอปพลิเคชันเชื่อมต่อหรือติดต่อกับฐานข้อมูล จะใช้คำสั่ง execSQL ดังตัวอย่างการใช้คำสั่ง ดังต่อไปนี้

- 1) การสร้างตาราง

ตัวอย่างคำสั่งการสร้างตาราง SQLite ดังนี้

```
CREATE TABLE table_name (_id INTEGER  
PRIMARY KEY AUTOINCREMENT, username TEXT,  
password TEXT, level INTEGER);
```

เมื่อนำมาใช้สร้างตารางบน Android จะเขียนดังภาพที่ 2.7

```

1 db.execSQL("CREATE TABLE table_name"
2           + " (_id INTEGER PRIMARY KEY
3             AUTOINCREMENT,"
4             + " username TEXT, password
              TEXT,"
              + " level INTEGER);");

```

ภาพที่ 2.7 คำสั่งสร้างตารางบน Android

2) การเพิ่มข้อมูล

คำสั่ง SQLite เพื่อเพิ่มข้อมูล ดังนี้

```

INSERT INTO table_name (username,
password, level)
VALUES ('akexorcist', '123456789', 50);

```

คำสั่งตาราง SQLite บน Android ดังภาพที่ 2.8

```

1 db.execSQL(INSERT INTO table_name
              (username, password, level)
2  VALUES ('akexorcist', '123456789',
              50);
3 );

```

ภาพที่ 2.8 คำสั่งสร้างตารางบน Android

2.3 ตัวอย่าง Gestures Builder Application

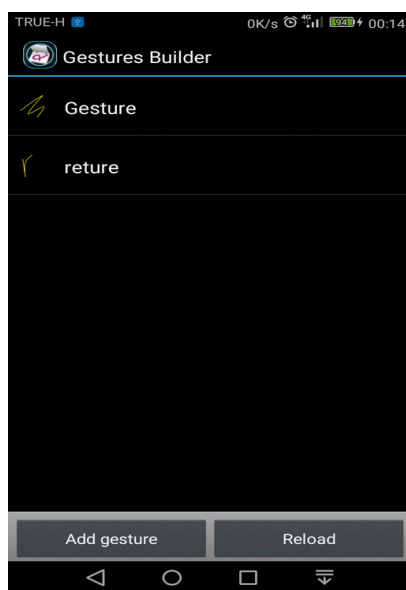
Gestures Builder Application เป็น Open Source ลิขสิทธิ์ (C) 2009 โครงการ Android Open Source ภายใต้สัญญาอนุญาต Apache เวอร์ชัน 2.9

Gestures Builder Application มีความสามารถดังต่อไปนี้

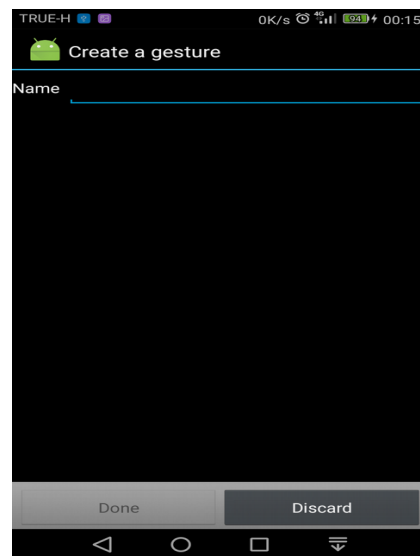
- 1) แสดงรายการ Gestures ที่ถูกบันทึกไว้ ดังภาพที่ 2.9

2) บันทึก Gestures เมื่อกดปุ่ม Add gesture จะเข้าไปที่หน้าให้บันทึก โดยจะมีช่องให้กรอกชื่อ และพื้นที่สำหรับวาด หากกดปุ่ม Done เป็นการยืนยันการบันทึก และปุ่ม Discard เป็นการยกเลิกการบันทึก ดังภาพที่ 2.10

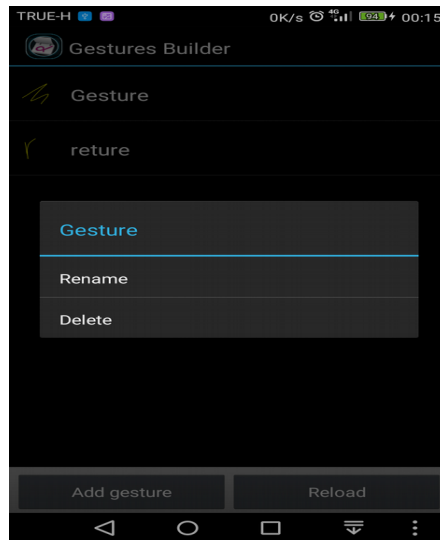
3) สามารถเปลี่ยนชื่อ Gestures ได้ด้วยการกดค้างที่ Gestures ที่ต้องการจะขึ้น Dialog ให้สามารถเปลี่ยนชื่อหรือลบ Gestures นั้น ดังภาพที่ 2.11



ภาพที่ 2.9 หน้าหลัก Gestures Builder



ภาพที่ 2.10 หน้าบันทึก Gestures



ภาพที่ 2.11 แสดง Dialog เปลี่ยนชื่อหรือลบ Gestures

บทที่ 3

การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และออกแบบระบบการสร้างสมมติฐานโครงการวิจัยด้วยเครื่องมือสนับสนุนการคิดเชิงระบบ มีขั้นตอนในการดำเนินงานต่อไปนี้

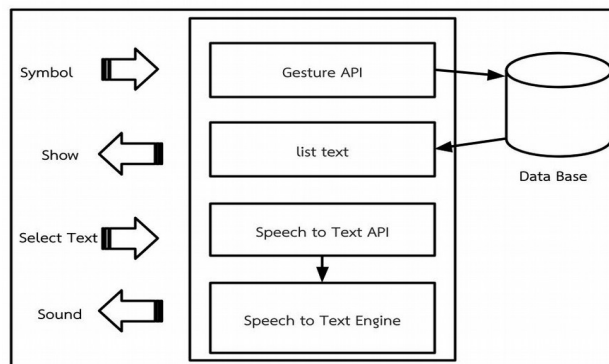
- 3.1 ภาพรวมของระบบ
- 3.2 Use case Diagram
- 3.3 Class Diagram
- 3.4 State Diagram
- 3.5 Sequence Diagram
- 3.6 Entity Relationship Diagram
- 3.7 User Interface Design)

3.1 ภาพรวมของระบบ

แอปพลิเคชันพูดสื่อใจ ประกอบไปด้วยการทำงาน 3 ส่วนหลัก ดังนี้

3.1.1 การแปลงข้อความเป็นเสียงสังเคราะห์

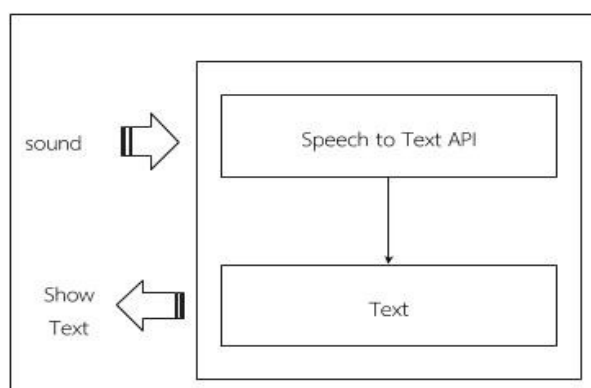
ส่วนของการทำงานในส่วนของการแปลงข้อความเป็นเสียงสังเคราะห์ ผู้ใช้ต้องวาดสัญลักษณ์ในพื้นที่ที่ระบุ จากนั้นระบบจะแสดงข้อความหรือประโยคที่แทนด้วยสัญลักษณ์นั้นเพื่อให้ผู้ใช้เลือก เมื่อผู้ใช้ทำการเลือกข้อความหรือประโยค แอปพลิเคชันจะแปลงข้อความหรือประโยคให้เป็นเสียงสังเคราะห์ ดังภาพที่ 3.1



ภาพที่ 3.1 ภาพรวมการทำงานของระบบการแปลงข้อความเป็นเสียงสังเคราะห์

3.1.2 การแปลงเสียงภายนอกเป็นข้อความ

ในการทำงานของการแปลงเสียงภายนอกเป็นข้อความ แอปพลิเคชันจะทำการแปลงเสียงที่รับเข้ามาแล้วแสดงข้อความตามเสียงที่รับเข้า ดังภาพที่

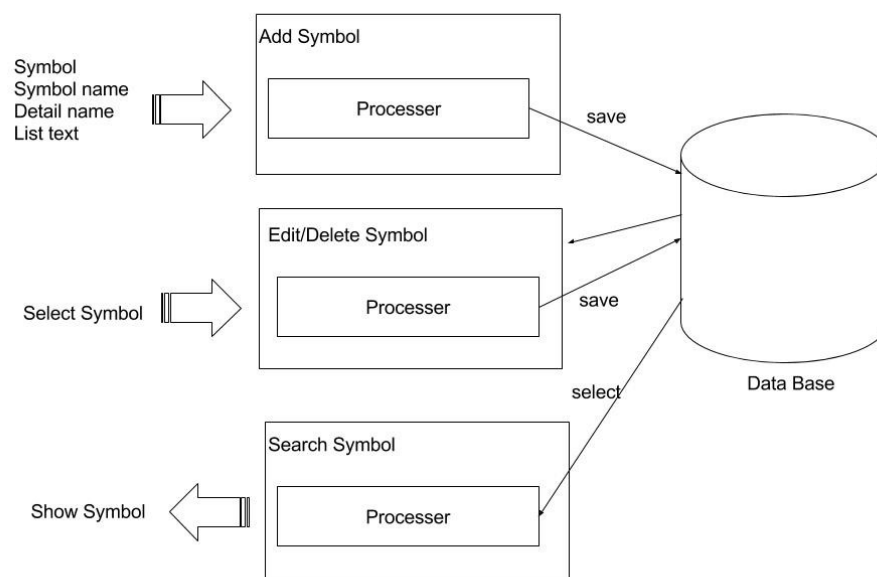


ภาพที่ 3.2 ภาพรวมการทำงานของระบบการแปลงข้อความภายนอกเป็นข้อความ

3.1.3 การจัดการสัญลักษณ์ในแอปพลิเคชันพุดสื่อใจ

การจัดการสัญลักษณ์ในแอปพลิเคชันพุดสื่อใจ เป็นส่วนที่เกี่ยวกับการจัดเก็บสัญลักษณ์ของแอปพลิเคชัน ซึ่งแบ่งการทำงานออก 3 ส่วน ได้แก่ การเพิ่มสัญลักษณ์ การค้นหาสัญลักษณ์ และการจัดการสัญลักษณ์ โดยมีรายละเอียด ดังต่อไปนี้

- 1) การเพิ่มสัญลักษณ์ เป็นการเก็บสัญลักษณ์แทนประโยคหรือสัญลักษณ์ ประกอบไปด้วย สัญลักษณ์ ชื่อแทนสัญลักษณ์ รายละเอียดคำอธิบายสัญลักษณ์ และรายการข้อความหรือประโยค
 - 2) การค้นหาสัญลักษณ์ เป็นการค้นหาสัญลักษณ์ที่มีอยู่ในแอปพลิเคชันจากการจัดเก็บของผู้ใช้
 - 3) จัดการสัญลักษณ์ เป็นการจัดการแก้ไขหรือลบสัญลักษณ์ในแอปพลิเคชัน
- ซึ่งการทำงานทั้ง 3 ส่วนมีภาพรวมการทำงาน ดังภาพที่

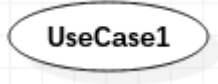
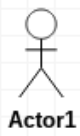


ภาพที่ 3.3 ภาพรวมการทำงานของจัดการสัญลักษณ์ในแอปพลิเคชันพุดสื่อใจ

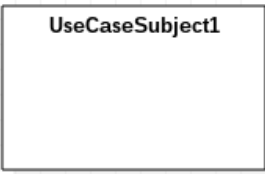


3.2 Use Case Diagram

Use Case Diagram เป็นโมเดลเพื่อแสดงฟังก์ชันการทำงานของระบบโดยรวมว่ามีส่วนประกอบใดบ้าง และมีกิจกรรมใดบ้างที่เกิดขึ้นในระบบ สัญลักษณ์ที่ใช้ในการเขียน Use Case Diagram ดังแสดงในตาราง

ตารางที่ 3.1 สัญลักษณ์ของ Use Case Diagram

สัญลักษณ์	คำอธิบาย
	User Case คือส่วนย่อยของระบบ แทนด้วยวงรีและชื่อ Use Case ในวงรี
	Actor คือบุคคลหรือระบบงานอื่นที่ใช้งานระบบหรือได้รับประโยชน์จากระบบซึ่งอยู่ภายนอกระบบ แทนด้วยรูปคนและมีชื่อบทบาทการใช้งานระบบ

ตารางที่ 3.1 สัญลักษณ์ของ Use Case Diagram (ต่อ)

สัญลักษณ์	คำอธิบาย
	กรอบสี่เหลี่ยมแสดงถึงขอบเขตของระบบโดยแสดงชื่อระบบภายในหรือด้านบนกรอบสี่เหลี่ยม Use Case อยู่ภายในกรอบสี่เหลี่ยม ส่วน Actor อยู่ภายนอกกรอบสี่เหลี่ยม
	ความสัมพันธ์แบบ <<User>> แสดงว่า Use Case หนึ่งดำเนินการตามขั้นตอนของ User Case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง Use Case ที่หัวลูกศรเรียกใช้งาน Use Case ที่หางลูกศรทุกครั้งที่มีการทำงาน
	ความสัมพันธ์แบบ <<extend>> แสดงว่า Use Case หนึ่งดำเนินการตามขั้นตอนของ Use Case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง

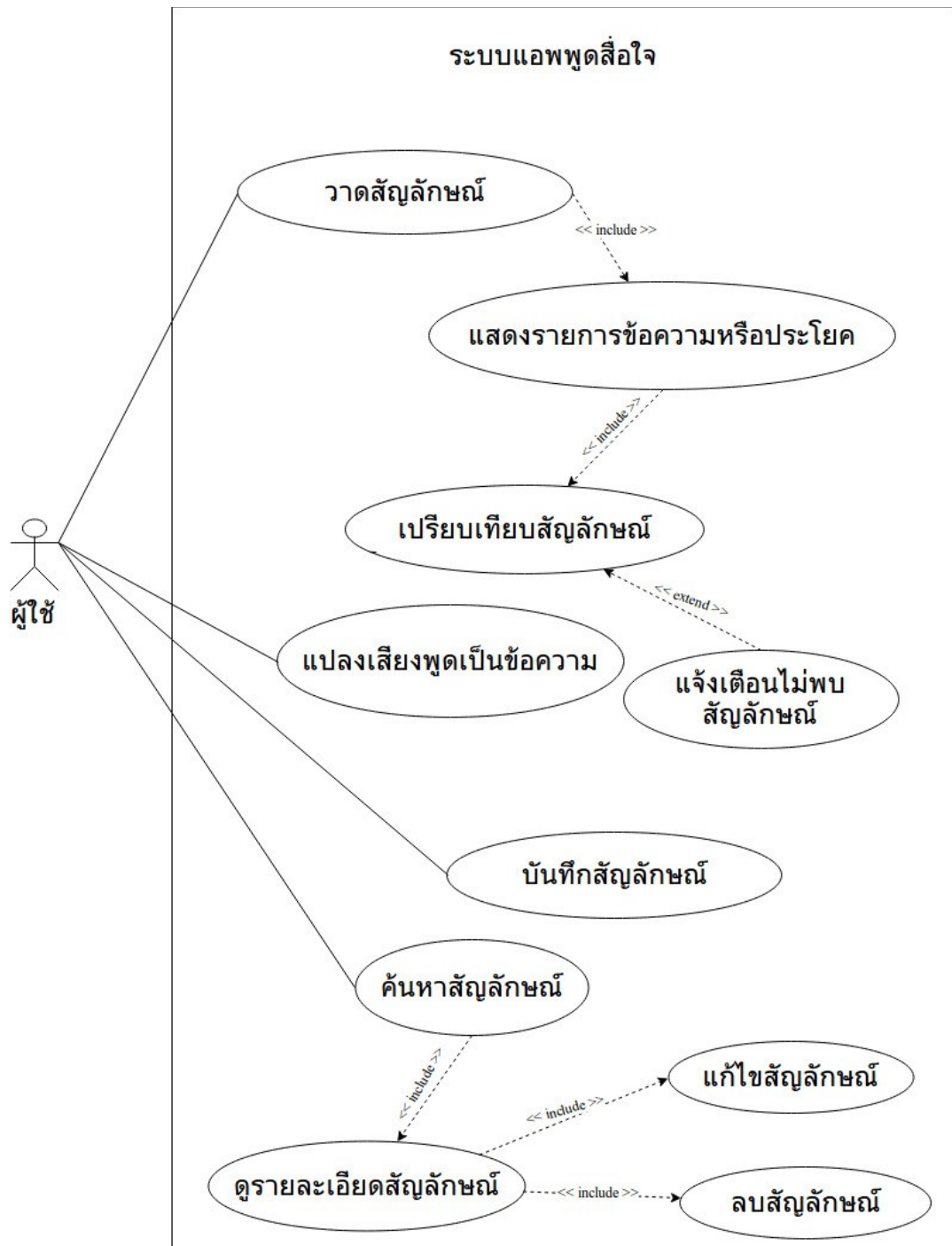
	Use Case ที่หัวหน้าลูกศรเรียกใช้งาน Use Case ที่หัวหน้าลูกศรเรียกใช้งาน Use Case ที่หางลูกศร แต่การใช้งานไม่จำเป็นต้องเกิดขึ้นทุกครั้งขึ้นอยู่กับเงื่อนไขระหว่างการทำงาน
--	--

ในการพัฒนาแอปพลิเคชันพุตสื่อใจ สามารถอธิบายขั้นตอนการทำงานที่สำคัญและความสัมพันธ์ต่าง ๆ ของระบบด้วย Use Case Diagram ดังแสดงในภาพที่ 3.4

ภาพที่ 3.4 Use Case Diagram แสดงการทำงานของแอปพลิเคชันพุตสื่อใจ

จากภาพที่ 3.4 ประกอบไปด้วย Use Case ดังนี้

- วาดสัญลักษณ์ : ทำหน้าที่ให้ผู้ใช้งานวาดสัญลักษณ์ในพื้นที่ที่ระบุ เพื่อเรียกใช้สัญลักษณ์
- แสดงรายการรายการข้อความหรือประโยค : ทำหน้าที่แสดงรายการข้อความหรือประโยค



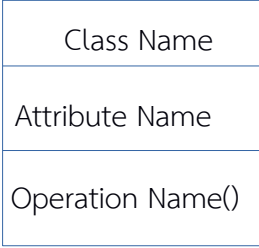
- เปรียบเทียบสัญลักษณ์ : ทำหน้าที่เปรียบเทียบหาสัญลักษณ์ที่มีอยู่ในแอปพลิเคชัน
- แจ้งเตือนไม่พบสัญลักษณ์ : ทำหน้าที่แจ้งเตือนเมื่อไม่พบสัญลักษณ์ในแอปพลิเคชัน
- แปลงเสียงพูดเป็นข้อความ : ทำหน้าที่แปลงเสียงที่รับเข้าเป็นข้อความ

- บันทึกสัญลักษณ์ : ทำหน้าที่เพิ่มสัญลักษณ์เข้าสู่แอปพลิเคชัน
- ค้นหาสัญลักษณ์ : ทำหน้าที่ช่วยค้นหาสัญลักษณ์ที่มีในแอปพลิเคชัน
- ดูรายละเอียดสัญลักษณ์ : ทำหน้าที่แสดงรายละเอียดทั้งหมดของแอปพลิเคชัน
- แก้ไขสัญลักษณ์ : ทำหน้าที่ไว้สำหรับแก้ไขเกี่ยวกับสัญลักษณ์ที่ต้องการ
- ลบสัญลักษณ์ : ทำหน้าที่ลบสัญลักษณ์ออกจาก Application


3.3 Class Diagram






Class Diagram คือแผนภาพที่ใช้แสดง Class และความสัมพันธ์ระหว่าง Class สัญลักษณ์ที่ใช้ในการเขียน Class Diagram แสดงดังตารางที่

ตารางที่ 3.2 สัญลักษณ์ของ Class Diagram

สัญลักษณ์	คำอธิบาย
	<p>คลาส สัญลักษณ์แทนด้วยสี่เหลี่ยมแบ่งเป็น 3 ส่วน ส่วนบนเป็นชื่อคลาส ส่วนกลางเป็น Attribute และส่วนล่างเป็น Operation Name หรือเมธอดที่เก็บรวบรวมข้อมูลที่แสดงถึงบุคคล สถานที่ และเหตุการณ์ที่มีความเกี่ยวข้องกับระบบ เมธอดเป็นการกระทำหรือฟังก์ชันที่คลาสนั้นสามารถทำได้ ชนิดของ Visibility ในเมธอด แบ่งเป็น 3 ชนิด ได้แก่</p>

ตารางที่ 3.2 สัญลักษณ์ของ Class Diagram (ต่อ)

สัญลักษณ์	คำอธิบาย
	<p>1) Public แทนสัญลักษณ์ด้วยเครื่องหมายบวก (+) 2) Private แทนสัญลักษณ์ด้วยเครื่องหมายลบ (-) 3) protected แทนสัญลักษณ์ด้วยเครื่องหมายชาร์ป (#)</p>
	<p>Dependency Relationship หมายความว่าคลาสที่อยู่ฝั่งต้นลูกศรเรียกใช้คลาสที่อยู่ฝั่งหัวลูกศร</p>

	Generalization หมายความว่าคลาสที่อยู่ฝั่งต้นลูกศร ทำการสืบทอดคลาสที่อยู่ฝั่งหัวลูกศร
	Association Relationship หมายความว่าคลาสที่อยู่ฝั่งต้นลูกศร ทำการกำหนดคลาสอื่นในรูป Attribute ภายในคลาส และสามารถเรียกใช้เมธอดจากคลาสนั้นได้
	Shared Aggregation หมายความว่า ความสัมพันธ์แบบเป็นส่วนหนึ่ง โดยจะมีคลาสที่อยู่ฝั่งหัวลูกศรเป็นคลาสหลัก และมีคลาสที่อยู่ฝั่งลูกศรเป็นส่วนหนึ่งของคลาส
	Composition Aggregation ความสัมพันธ์แบบขึ้นต่อกันและเกี่ยวข้องกันเสมอ
	Connector เป็นสัญลักษณ์แทนด้วยรูปหน้าเหลี่ยม และมีชื่ออยู่กลาง จะสร้างสัญลักษณ์นี้ไว้ ต้องการเชื่อมต่อคลาสที่อยู่คนละหน้า

Class Diagram แสดงความสัมพันธ์ในรูปแบบต่าง ๆ ระหว่างคลาสของแอปพลิเคชันพุตสื่อ
ใจ อธิบายตามภาพที่

ภาพที่ 3.5 ภาพ Class Diagram ของแอปพลิเคชันพุตสื่อใจ

MEMO_gestureManagementFragment	คลาสหน้าระบุสัญลักษณ์เพื่อจัดเก็บในฐานข้อมูล
MEMO_listviewAdapter	คลาสสำหรับให้คลาสอื่นกำหนดเป็น Attribute ภายในคลาส และเรียกใช้ Method ได้
MEMO_mainFragment	คลาสหน้าแสดงหลักของส่วนบันทึกแอปพลิเคชันพูดสื่อใจ และการค้นหาสัญลักษณ์
MEMO_managementFragment	คลาสแบบฟอร์มเก็บรายละเอียดสัญลักษณ์เพื่อจัดเก็บในฐานข้อมูล
MyGesture	คลาส Model ของสัญลักษณ์
MyTTS	คลาสจัดการ Text to Speech
SST_mainFragment	คลาสแสดงหน้าหลักและการทำงานของการเล่นเสียงเป็นข้อความหรือประโยค
TTS_groupListFragment	คลาสแสดงรายการข้อความหรือประโยคเพื่อเลือกแล้วแปลงเป็นเสียงสังเคราะห์
TTS_mainFragment	คลาสหน้าหลักของการแปลงข้อความเสียง แสดงพื้นที่ให้ระบุสัญลักษณ์เพื่อดีงรายการข้อความหรือประโยคที่แทนด้วยสัญลักษณ์นั้น

ตารางที่ 3.3 อธิบายแผนภาพ Class Diagram ของภาพที่ 3.5 (ต่อ)






คลาส	คำอธิบาย
TTS_textShowFragment	คลาสแสดงข้อความหรือประโยคที่ถูกเลือกและแปลงเป็นเสียงสังเคราะห์

3.4 Sequence Diagram

Sequence Diagram เป็น Diagram ที่แสดงขั้นตอนการทำงานของแต่ละ Use Case ระหว่าง Object ต่าง ๆ ที่ส่งข้อความถึงกันและกัน โดย Sequence diagram จะช่วยให้มองเห็นการ

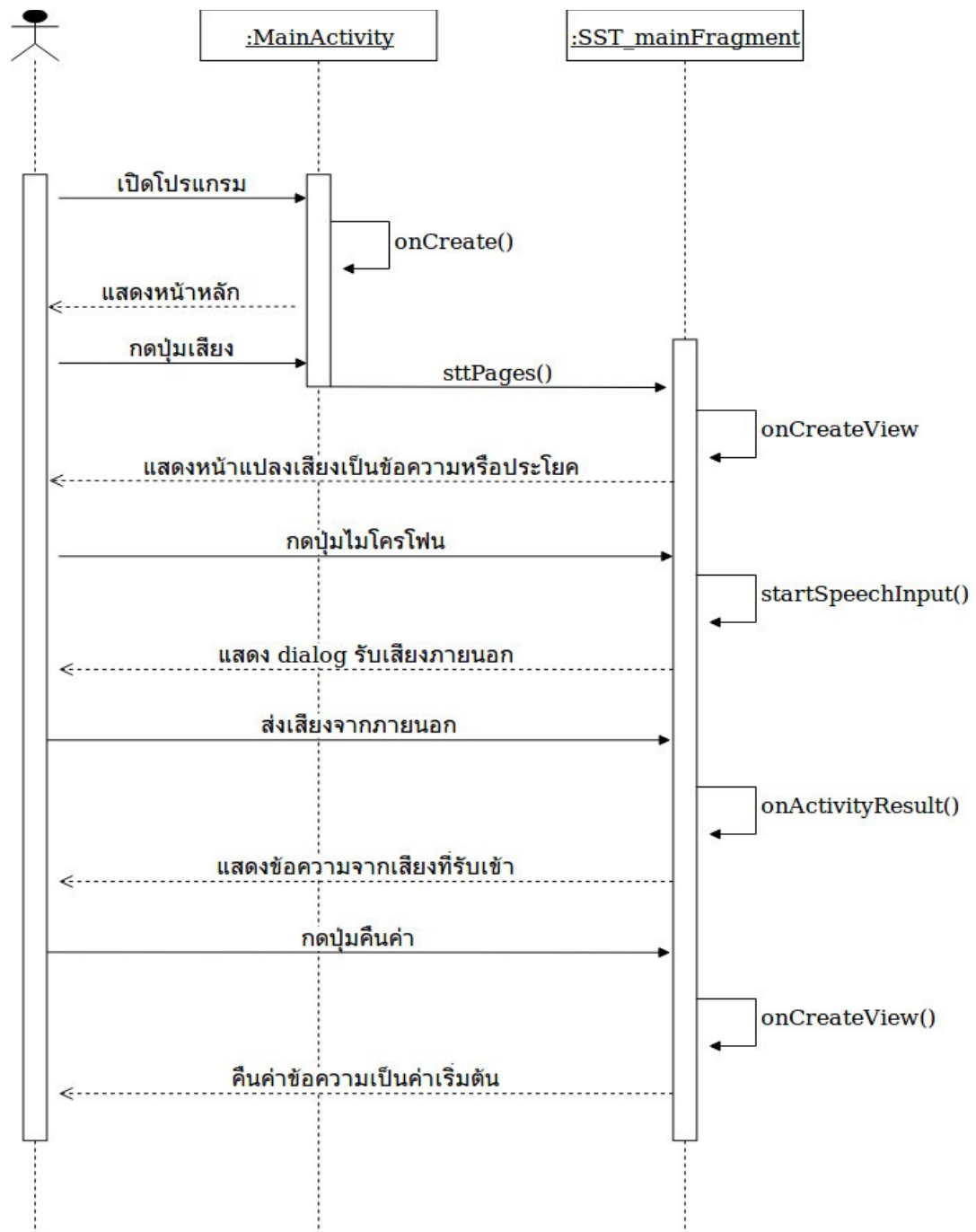
ทำานของภาพรวมจองระบบ ส่วนประกอบและสัญลักษณ์ที่ใช้ในการเขียน Sequence Diagram ดังแสดงในตารางที่

ตารางที่ 3.4 สัญลักษณ์ของ Sequence Diagram

สัญลักษณ์	การใช้งาน
	Class แสดงถึงการทำงานของ Use Case ในการส่งหรือรับ Message สัญลักษณ์แทนด้วยสี่เหลี่ยมมีชื่อคลาสอยู่ภายใน
	Lifeline หรือเส้นอายุขัย แสดงช่วงเวลาตั้งแต่เริ่มสร้าง object ใน class นั้น จนกระทั่ง object นั้น ถูกทำลาย สัญลักษณ์แทนด้วยเส้นประ
	Focus of control หรือจุดควบคุม เป็นจุดควบคุมที่ Object ใช้ทำการส่งหรือรับ Message สัญลักษณ์แทนด้วยสี่เหลี่ยม
	Message คือ ข้อความที่รับส่งระหว่าง Object สัญลักษณ์แทนด้วยลูกศรและประกอบด้วย 2 ส่วน คือ ข้อมูล (Data) : Function
	Return Message เป็นข้อมูลที่ส่งกลับหลังจากทำงานเสร็จ

Sequence Diagram ที่สำคัญของแอปพลิเคชันพุตสื่อใจ มีรายละเอียดดังต่อไปนี้

ภาพที่ 3.6 Sequence Diagram การแปลงเสียงภายนอกเป็นข้อความ



จากภาพที่ 3.6 สามารถอธิบายแผนภาพ Sequence Diagram การแปลงเสียงเป็นข้อความ ได้ดังต่อไปนี้ เมื่อผู้ใช้เปิดโปรแกรมเข้ามาระบบจะเรียก Method ชื่อ onCreate ใน Class ชื่อ


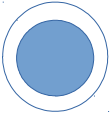



MainActivity เพื่อสร้างหน้าจอหลักของโปรแกรม เมื่อผู้ใช้เรียกกดปุ่มเสียงในแถบเมนู Class ชื่อ MainActivity จะเรียกใช้ Method ชื่อ onClick เพื่อเรียกใช้งาน Class ชื่อ SST_mainFragment แล้วเรียกใช้งาน Method ชื่อ onCreateView เพื่อสร้างหน้าแปลงเสียงเป็นข้อความมาแสดง เมื่อผู้ใช้กดปุ่มรูปไมล์ Method ชื่อ startActivityInput() จะถูกเรียกใช้งาน จากนั้นแสดง dialog เพื่อบอกให้ผู้ใช้ส่งเสียงเข้าแอปพลิเคชัน ต่อมา Class ชื่อ SST_mainFragment เรียกใช้ Method ชื่อ onActivityResult เพื่อแสดงข้อความจากเสียงที่รับเข้า หากผู้ใช้กดปุ่มค้นหา จะเรียกใช้งาน Class ชื่อ onCreate และทำการค้นหาข้อความเพื่อแสดงข้อความเป็นค่าเริ่มต้น

ภาพที่ 3.7 Sequence Diagram การเพิ่มสัญลักษณ์

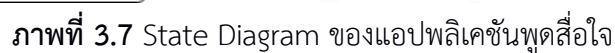


State Diagram เป็น diagram ที่แสดงสถานะของ object แต่ละตัว โดยสถานะโดยรวมของระบบเกิดจากสถานะย่อยของ Object แต่ละตัวรวมกัน กลไกที่ทำให้ระบบมีการเปลี่ยนสถานะ คือ การส่ง message ในทาง object orientation ซึ่งก็คือ การเรียกใช้ function ของ object นั้นเอง ส่วนประกอบและสัญลักษณ์ที่ใช้ในการเขียน State Diagram ดังแสดงในตารางที่

ตารางที่ 3.5 สัญลักษณ์ของ State Diagram

สัญลักษณ์	คำอธิบาย
	Initial state คือสถานะเริ่มต้นแสดงถึง object ที่เกิดขึ้น
	Final state คือสถานะสิ้นสุด แสดงสถานะที่ object ทำงานเสร็จสิ้น
	State คือแสดงสถานะ object
	Event คือเหตุการณ์ที่เกิดขึ้น ทำให้เกิดการเปลี่ยนสถานะ โดยมีเงื่อนไข ซึ่ง object จะเปลี่ยนสถานะเมื่อเงื่อนไขดังกล่าวเป็นจริง
	Transition คือการเปลี่ยนสถานะแสดงถึงการเปลี่ยนสถานะของ object จากสถานะหนึ่งไปยังสถานะอื่น

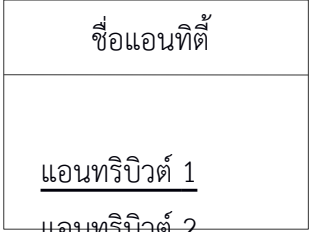

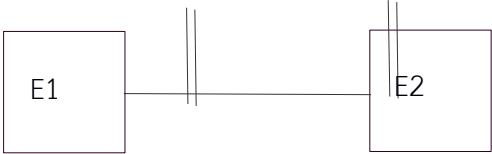
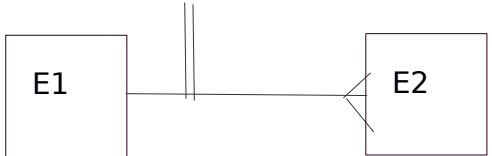
State Diagram แสดงการทำงานของแอปพลิเคชันพุดสื่อใจ มีรายละเอียดดังนี้



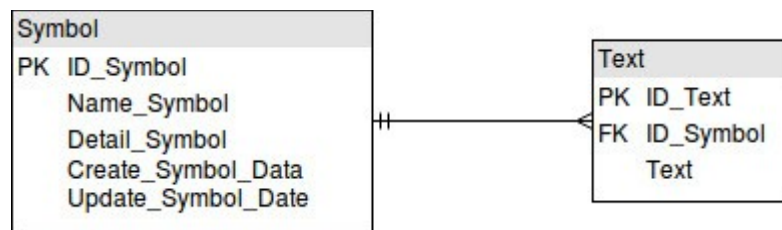
3.6 Entity Relationship Diagram

แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล ER Diagram (Entity Relationship Diagram) คือแผนภาพที่ใช้เป็นเครื่องมือสำหรับจำลองข้อมูล ซึ่งประกอบไปด้วย Entity คือ กลุ่มของข้อมูลและความสัมพันธ์ มีหลายชนิด เช่น ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One), หนึ่งต่อหลาย (One to Many), หรือ หลายสิ่งต่อหลายสิ่ง (Many to Many)

ตารางที่ 3.6 สัญลักษณ์ของ ER Diagram

สัญลักษณ์	ความหมาย
	<p>Entity ปกติ และ Attribute จะแสดงข้างใต้ชื่อ Entity โดย Attribute ที่เป็นคีย์หลัก จะขีดเส้นใต้ชื่อ Attribute</p> <p>PK คือ Primary Key</p> <p>FK คือ Foreign Key</p>
	<p>ความสัมพันธ์ระหว่างแอนทิดีปกติชื่อ ความสัมพันธ์</p>
	<p>ความสัมพันธ์ของข้อมูลระหว่าง E1 กับ E2 แบบ 1:1 แบบ Total Participation (Mandatory)</p>
	<p>ความสัมพันธ์ของข้อมูลระหว่าง E1 กับ E2 แบบ 1:N โดย แบบ E1 เป็นแบบ Participation (Optional) ส่วน E2 เป็นแบบ Total Participation (Mandatory)</p>

ER Diagram ของแอปพลิเคชันพุตสือใจ มีรายละเอียดดังนี้



ภาพที่ 3.8 ER Diagram ของแอปพลิเคชันพุดสื่อใจ

พจนานุกรมข้อมูล ของแอปพลิเคชันพุดสื่อใจ ประกอบด้วยพจนานุกรมทั้งหมด 2 ตารางดังนี้

ตารางที่ 3.7 ตารางสัญลักษณ์

ชื่อ	ชนิด	สถานะ	คำอธิบาย
ID_Symbol	Int	PK	รหัสสัญลักษณ์
Name_Symbol	Text		ชื่อแทนสัญลักษณ์
Detail_Symbol	Text		รายละเอียดคำอธิบายสัญลักษณ์
Create_Symbol_Date	Datetime		วันที่สร้างสัญลักษณ์
Update_Symbol_Date	Datetime		วันอัปเดตสัญลักษณ์

ตารางที่ 3.8 ตารางข้อความหรือประโยค

ชื่อ	ชนิด	สถานะ	คำอธิบาย
ID_Text	Int	PK	รหัสข้อความหรือประโยค
Text	Text		ข้อความหรือประโยค
ID_Symbol	Int	FK	รหัสสัญลักษณ์

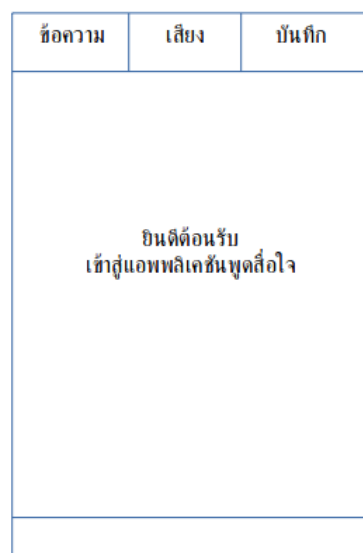
3.7 User Interface Design

ในการออกแบบ User interface Design ของแอปพลิเคชันพุดสื่อใจ มีรายละเอียด ดังนี้

3.7.1 การออกแบบหน้าหลักของแอปพลิเคชันพุดสื่อใจ

แสดงหน้าแรกของแอปพลิเคชันพุดสื่อใจ จะเห็นว่ามี 3 ส่วนหลัก คือ ข้อความ เสียง และ บันทึก ผู้ใช้สามารถเข้าไปที่ส่วนนั้นได้ โดยการกดเข้าไปที่ปุ่มนั้น ดังภาพที่ 3.9 อธิบายแต่ละเมนูดังนี้

- ข้อความ คือ การแปลงข้อความเป็นเสียงสังเคราะห์
- เสียง คือ การแปลงเสียงภายนอกเป็นข้อความ
- บันทึก คือการจัดการสัญลักษณ์ในแอปพลิเคชันพุดสื่อใจ



ภาพที่ 3.9 ภาพหน้าแรกของแอปพลิเคชันพุดสื่อใจ

3.7.2 การออกแบบส่วนข้อความ

1) การออกแบบหน้าแรกของข้อความ

ภาพหน้าแรกของข้อความ จะมีพื้นที่ให้ผู้ใช้งานเพื่อบันทึกสัญลักษณ์แทนรายการข้อความหรือประโยค และกดยืนยันเพื่อแสดงรายการข้อความหรือประโยคที่ถูกแทนด้วยสัญลักษณ์นั้น ดังภาพที่ 3.10

ภาพที่ 3.10 ภาพหน้าแรกของข้อความ

ข้อความ	เสียง	บันทึก
<p>กรุณาลากนิ้วเป็นสัญลักษณ์บนหน้าจอ</p> <p>ยืนยัน</p>		

2) การออกแบบหน้าแสดงรายการข้อความหรือประโยค

ผู้ใช้สามารถเลือกข้อความหรือประโยคจากรายการเพื่อแสดงเสียงสังเคราะห์จากข้อความหรือประโยคนั้น ดังภาพที่ 3.11

ข้อความ	เสียง	บันทึก
ข้อความ หรือ ประโยค 1		
ข้อความ หรือ ประโยค 2		
ข้อความ หรือ ประโยค 3		
ข้อความ หรือ ประโยค 4		
ข้อความ หรือ ประโยค 5		
ข้อความ หรือ ประโยค 6		
ข้อความ หรือ ประโยค 7		
ข้อความ หรือ ประโยค 8		
ข้อความ หรือ ประโยค 9		
ข้อความ หรือ ประโยค 10		
ข้อความ หรือ ประโยค 11		
ข้อความ หรือ ประโยค 12		

ภาพที่ 3.11 ภาพแสดงรายการข้อความหรือประโยค

3) การออกแบบหน้าแสดงข้อความหรือประโยค

การออกแบบหน้าแสดงข้อความหรือประโยค มีการออกแบบดังภาพที่ 3.12

ภาพที่ 3.12 ภาพหน้าแสดงข้อความหรือประโยค

ข้อความ	เสียง	บันทึก
<div style="border: 1px solid black; height: 150px; position: relative;"> <div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);"> แสดงข้อความ หรือประโยค </div> </div>		
<div style="display: flex; justify-content: space-around;"> <div>อีกครั้ง</div> <div>โต้ตอบ</div> </div>		

ในหน้าี้จะมีพื้นที่แสดงข้อความหรือประโยคที่เลือกมาจากหน้ารายการแสดงข้อความหรือประโยคที่ต้องการ มีปุ่มอีกครั้งเพื่อแสดงเสียงสังเคราะห์จากข้อความหรือประโยคนั้น และอีกปุ่มคือปุ่มโต้ตอบเพื่อเข้าไปที่ส่วนเสียง

3.3.3 การออกแบบส่วนเสียง

ข้อความ	เสียง	บันทึก
<div style="border: 1px solid black; height: 150px; position: relative;"> <div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%);"> แสดงข้อความ </div> </div>		
<div style="display: flex; justify-content: space-around;"> <div>ลบ</div> <div>พูด</div> </div>		




ภาพที่ 3.13 ภาพส่วนแสดงข้อความจากเสียง

จากภาพที่ 3.13 จะเห็นว่ามีปุ่ม 2 ปุ่ม คือ ปุ่มพูดและปุ่มเสียง เมื่อผู้ใช้กดที่ปุ่มพูดจะสามารถรับค่าเสียงเข้าไปแสดงที่พื้นที่แสดงข้อความ และเมื่อกดที่ปุ่มลบข้อความก่อนหน้านี้จะหายไป

3.3.4 การออกแบบส่วนบันทึก

1) การออกแบบหน้าแรกบันทึก

การออกแบบหน้าแรกบันทึก ออกแบบให้แสดงรายการสัญลักษณ์ที่เคยถูกบันทึกทั้งหมดในฐานข้อมูล โดยสามารถค้นหาได้ ดังภาพที่ 3.14

ข้อความ	เสียง	บันทึก
<input type="text"/>		
ค้นหา		
	ชื่อแทนสัญลักษณ์ รายละเอียดสัญลักษณ์	
	ชื่อแทนสัญลักษณ์ รายละเอียดสัญลักษณ์	
	ชื่อแทนสัญลักษณ์ รายละเอียดสัญลักษณ์	
		+

ภาพที่ 3.14 ภาพแสดงหน้าแรกส่วนบันทึกแสดงรายการสัญลักษณ์

2) การออกแบบหน้าบันทึกสัญลักษณ์

การออกแบบหน้าบันทึกสัญลักษณ์ จะมีหน้าต่อเนื่องกันสองหน้าโดยหน้าแรก 3.15 เป็นหน้าสำหรับกรอกข้อมูลเกี่ยวกับสัญลักษณ์ได้แก่ ชื่อสัญลักษณ์ รายละเอียดสัญลักษณ์ และข้อความหรือประโยคโดยมีปุ่มเพิ่มข้อความหรือสัญลักษณ์ ปุ่มถัดไปเพื่อไปหน้าถัดไป หน้าถัดมาเป็นหน้าสำหรับวาดสัญลักษณ์มีปุ่มย้อนกลับไปหน้าก่อนหน้าและปุ่มบันทึกเพื่อบันทึกสัญลักษณ์ใหม่เข้าไปในฐานข้อมูลดังภาพที่ 3.16

ข้อความ	เสียง	บันทึก
ชื่อ _____		
<div style="border: 1px solid black; height: 40px; width: 100%;"></div>		
<input type="button" value="เพิ่ม"/>		
<div style="border: 1px solid black; height: 100px; width: 100%; position: relative;"> X </div>		
<input type="button" value="ถัดไป"/>		

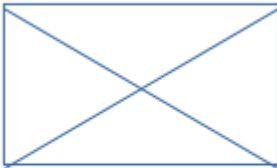
ภาพที่ 3.15 หน้ากรอกข้อมูลเกี่ยวกับสัญลักษณ์

ข้อความ	เสียง	บันทึก
<div style="border: 1px solid black; height: 150px; width: 100%; position: relative;"> X </div>		
<input type="button" value="ย้อนกลับ"/> <input type="button" value="ถัดไป"/>		

ภาพที่ 3.16 หน้าวาดสัญลักษณ์

3) การออกแบบหน้าแสดงรายละเอียดสัญลักษณ์

การออกแบบหน้าแสดงรายละเอียดสัญลักษณ์ จะแสดงข้อมูลเกี่ยวกับสัญลักษณ์ ได้แก่ ชื่อสัญลักษณ์ รายละเอียดสัญลักษณ์ ภาพสัญลักษณ์ และรายการข้อความหรือประโยค มีปุ่มสำหรับแก้ไขข้อมูลสัญลักษณ์ ปุ่มลบสัญลักษณ์ออกจากรฐานข้อมูล ดังภาพที่ 3.17

ข้อความ	เสียง	บันทึก
ชื่อ <input type="text"/> <input type="text"/>		
		
<hr/> <hr/> <p>แสดงรายการข้อความ</p> <hr/> <hr/>		

ภาพที่ 3.17 หน้าแสดงรายละเอียดสัญลักษณ์

บทที่ 4

การพัฒนาระบบงาน

ในบทนี้จะกล่าวถึงการสร้างระบบงานของแอปพลิเคชันพูดสื่อใจ โดยนำผลที่ได้จากการวิเคราะห์และออกแบบระบบมาสร้างเป็นระบบงาน ซึ่งจะอธิบายถึงตัวอย่างการเขียนโปรแกรมการทำงานของระบบในส่วนต่าง ๆ ดังต่อไปนี้

4.1 การแปลงเสียงเป็นข้อความ

4.2 การแปลงข้อความเป็นเสียง

4.1 การแปลงเสียงเป็นข้อความ

การแปลงเสียงเป็นข้อความมีการ import android.speech.RecognizerIntent มาช่วยในการเรียกใช้งาน โดยเรียกใช้งานด้วย Intent และคำสั่ง startActivityForResult ดังแสดงในภาพที่

```
1 public class SST_mainFragment extends Fragment {  
2     private TextView voiceText;  
3     private android.support.design.widget.FloatingActionButton talkButton;  
4     private android.support.design.widget.FloatingActionButton resetButton;  
5     private final int REQUEST_VOIC_RECOGNITION = 10101;  
6     @Override  
7     public View onCreateView(LayoutInflater inflater, ViewGroup container,  
8                             Bundle savedInstanceState) {  
9         return inflater.inflate(R.layout.fragment_sst_main, container, false);  
10    }  
11    @Override  
12    public void onViewCreated(View view, @Nullable Bundle
```

```

        savedInstanceState) {
13         super.onCreate(view, savedInstanceState);
14         voiceText = (TextView) view.findViewById(R.id.textShow);
15         talkButton = (android.support.design.widget.FloatingActionButton)
16             view.findViewById(R.id.talk_Button);
17         talkButton.setOnClickListener(new View.OnClickListener() {
18             @Override
19             public void onClick(View v) {
20                 startSpeechInput();
21             }
22         });
23         resetButton = (android.support.design.widget.FloatingActionButton)
24             view.findViewById(R.id.reset_Button);
25         resetButton.setOnClickListener(new View.OnClickListener(){
26             @Override
27             public void onClick(View v) {
28                 voiceText.setText("Show Text");
29             }
30         });
31     }
32     public void startSpeechInput(){
33         Intent intent = new
34             Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
35         intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, "th-TH");
36         intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
37             RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
38         intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "พูดที่นี่");
39         try{
39             startActivityForResult(intent, REQUEST_VOIC_RECOGNITION);

```

ภาพที่ การแปลงเสียงเป็นข้อความ

บรรทัดที่ 2-4 เป็นการประกาศตัวแปรเพื่อใช้กำหนดค่าให้หน้า layout

บรรทัดที่ 5 ประกาศตัวแปรค่าคงที่ชื่อ REQUEST_VOIC_REQUEST_VOIC_RECOGINITION มีค่าเท่ากับ 10101

บรรทัดที่ 9 กำหนดให้ SST_mainFragment เรียก layout ชื่อ fragment_sst_main.xml มาแสดง

บรรทัดที่ 14 กำหนด id ชื่อว่า textShow ให้กับตัวแปรชื่อ voiceText

บรรทัดที่ 15 กำหนด id ชื่อว่า talk_Button ให้กับตัวแปรชื่อ talkButton

บรรทัดที่ 17-22 กำหนดให้หลังจากกดปุ่มที่มีตัวแปรชื่อ talkButton จะไปเรียกใช้งาน method ชื่อ startSpeechInput

บรรทัดที่ 23-24 กำหนด id ชื่อว่า reset_Button ให้กับตัวแปรชื่อ resetButton

บรรทัดที่ 25-30 กำหนดให้หลังจากกดปุ่มที่มีตัวแปรชื่อ resetButton จะไปตั้งค่าข้อความเป็นข้อความว่างให้ตัวแปรชื่อ voiceText

บรรทัดที่ 33 ประกาศ Intent ตัวแปรชื่อ intent เพื่อเรียกใช้งาน Voice Recognition ของ Android

บรรทัดที่ 34-35 เป็นการกำหนดภาษาที่รับข้อความเสียงเป็นภาษาไทย

บรรทัดที่ 37 ตั้งค่าข้อความบน Dialog ของ Voice Recognition เป็น “พูดที่นี่”

บรรทัดที่ 38-41 ส่งค่าให้ตัวแปรชื่อ intent และตัวแปร REQUEST_VOIC_RECOGINITION ไปที่ method ชื่อ startActivityForResultResult เพื่อใช้งาน

บรรทัดที่ 46-52 เป็นการกำหนดค่าข้อความให้ตัวแปรชื่อ voiceText ที่ได้จากผลลัพธ์ของคำสั่ง RecognizerInternt โดยมีเงื่อนไขว่าตัวแปรชื่อ requestCode มีค่าเท่ากับตัวแปร

REQUEST_VOIC_RECOGNITION ตัวแปรชื่อ resultCode เท่ากับ RESULT_OK และตัวแปรชื่อ data ไม่เป็นค่าว่าง

4.2 การแปลงข้อความเป็นเสียง

การแปลงข้อความเป็นเสียงในการพัฒนาส่วนนี้ได้ import android.speech.tts.TextToSpeech มาช่วยในการพัฒนา โดยแสดงดังภาพที่

```
1 public class MyTTS extends UtteranceProgressListener
2             implements TextToSpeech.OnInitListener,
3
4             TextToSpeech.OnUtteranceCompletedListener {
5     public static MyTTS myTTS;
6     public static MyTTS getInstance(Context context) {
7         if (myTTS == null) {
8             myTTS = new MyTTS(context);
9         }
10        return myTTS;
11    }
12    private Context context;
13    private TextToSpeech tts;
14    private Locale locale = Locale.getDefault();
15    private String enginePackageName;
16    private String message;
17    private boolean isRunning = false;
18    private int speakCount;
19    public MyTTS(Context context) {
```

```
19     this.context = context;
20 }
21 public void speak(String message) {
22     this.message = message;
23     if (tts == null || !isRunning) {
24         speakCount = 0;
25         if (enginePackageName != null && !enginePackageName.isEmpty()) {
26             tts = new TextToSpeech(context, this, enginePackageName);
27         } else {
28             tts = new TextToSpeech(context, this);
29         }
30         if (Build.VERSION.SDK_INT >=
            Build.VERSION_CODES.ICE_CREAM_SANDWICH_MR1) {
31             tts.setOnUtteranceProgressListener(this);
32         } else {
33             tts.setOnUtteranceCompletedListener(this);
34         }
35         isRunning = true;
36     } else {
37         startSpeak();
38     }
39 }
40 ...
41 private void startSpeak() {
42     speakCount++;
43     if (locale != null) {
44         tts.setLanguage(locale);
45     }
```

```

46     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
47         tts.speak(message, TextToSpeech.QUEUE_FLUSH, null, "");
48     } else {
49         tts.speak(message, TextToSpeech.QUEUE_FLUSH, null);
50     }
51 }
52 private void clear() {
53     speakCount--;
54     if (speakCount == 0) {
55         tts.shutdown();
56         isRunning = false;
57     }
58 }
59 ...

```

ภาพที่ การแปลงข้อความเป็นเสียง

บรรทัดที่ 5-10 เป็นการประกาศ Method ชื่อ getInstance เพื่อกำหนดค่า Context ให้ Class ชื่อ MyTTS เมื่อตัวแปรชื่อ myTTS มีค่าว่าง

บรรทัดที่ 11-17 เป็นการประกาศตัวแปรเพื่อใช้ภายใน Class ชื่อ MyTTS

บรรทัดที่ 18-20 เป็น Constructor ของ Class คือ MyTTS

บรรทัดที่ 21 ประกาศ Method ชื่อ speak รับค่าเข้าเป็น String ชื่อ message

บรรทัดที่ 22 กำหนดให้ค่าตัวแปรชื่อ message ใน Class ชื่อ MyTTS มีค่าเท่ากับ ตัวแปรชื่อ message ที่รับเข้ามาใน Method ชื่อ speak

บรรทัดที่ 23-26 ถ้าตัวแปรชื่อ tts เป็นค่าว่างหรือตัวแปรชื่อ isRunning มีค่าเป็นเท็จ ให้ กำหนดค่าตัวแปรชื่อ speakCount มีค่าเท่ากับ 0 และถ้าตัวแปรชื่อ enginePackageName ไม่เป็น ค่าว่าง กำหนดให้ตัวแปรชื่อ tts ส่งค่าตัวแปร ได้แก่ context, this และ enginePackageName ไป

ที่ Class ชื่อ TextToSpeech แต่ถ้าว่างส่งค่าตัวแปร ได้แก่ context และ enginePackageName ไปที่ Class ชื่อ TextToSpeech

บรรทัดที่ 27-29 ถ้าไม่เข้าเงื่อนไขตัวแปรชื่อ tts เป็นค่าว่างหรือตัวแปรชื่อ isRunning มีค่าเป็นเท็จ ให้ส่งค่าตัวแปร ได้แก่ context และ enginePackageName ไปที่ Class ชื่อ TextToSpeech

บรรทัดที่ 30-34 ถ้า SDK Version มากกว่าหรือเท่ากับ ICE_CREAM_SANDWICH_MR1 ให้ตัวแปรชื่อ tts เรียกใช้ Method ชื่อ setOnUttranceProgressListener แต่ถ้าไม่เข้าเงื่อนไขให้ตัวแปรชื่อ tts เรียกใช้ Method ชื่อ setOnUtteranceCompletedListener

บรรทัดที่ 35 กำหนดค่าให้ตัวแปรชื่อ isRunning มีค่าเท่ากับ true

บรรทัดที่ 36-38 ถ้าไม่เข้าเงื่อนไขในบรรทัดที่ 23 เรียกใช้ Method ชื่อ startSpeak

บรรทัดที่ 41 สร้าง Method ชื่อว่า startSpeak

บรรทัดที่ 42 ตัวแปรชื่อ speakCount มีค่าเพิ่มขึ้นทีละหนึ่ง

บรรทัดที่ 43-44 เป็นการกำหนดตัวแปรชื่อ locale ไปที่ Method setLanguage เพื่อกำหนดภาษาถ้าตัวแปรชื่อ locale มีค่าว่าง

บรรทัดที่ 46-50 เป็นการแปลงข้อความเป็นเสียงพูดโดยใช้คำสั่ง speak

บรรทัดที่ 52-58 Method ชื่อว่า clear ถ้าตัวแปรชื่อ speakCount มีค่าเท่ากับ 0 ให้ตัวแปรชื่อ tts เรียกการใช้งาน Method ชื่อ shutdown และกำหนดค่าให้ตัวแปรชื่อ isRunnig เป็น false

บทที่ 5

การทดสอบระบบ

ในบทนี้ การทดสอบระบบเป็นการทดสอบการทำงานของแอปพลิเคชันพุตสื่อใจ แบ่งการทดสอบออกเป็น 4 ส่วน ดังนี้

- 5.1 การทดสอบส่วนแปลงเสียงภายนอกเป็นข้อความ
- 5.2 การทดสอบส่วนบันทึกสัญลักษณ์แทนข้อความหรือประโยค
- 5.3 การทดสอบส่วนแปลงข้อความเป็นเสียงสังเคราะห์

5.1 การแปลงเสียงภายนอกเป็นข้อความ มีการทดสอบดังนี้

ตารางที่ การทดสอบแปลงเสียงภายนอกเป็นข้อความ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
แปลงเสียงภายนอกเป็นข้อความ	กดปุ่มรูปไมโครโฟน	แสดง Dialog สำหรับรับค่าเสียง
	ส่งเสียงเข้าสู่แอปพลิเคชัน	แสดงข้อความตามเสียงที่ได้รับ
	กดปุ่มรูปกากบาท	คืนค่าข้อความ

5.2 การบันทึกสัญลักษณ์แทนข้อความหรือประโยค มีการทดสอบดังนี้

ตารางที่ การทดสอบแปลงเสียงภายนอกเป็นข้อความ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
บันทึกสัญลักษณ์	กดปุ่ม เพิ่ม สัญลักษณ์	แสดงหน้ากรอกข้อมูลสัญลักษณ์
	กรอกข้อมูล	ไม่มีการดักข้อมูล
	กดปุ่ม เพิ่ม	แสดงข้อความหรือประโยคที่เพิ่มเข้า
	กดปุ่มถัดไป	แสดงหน้าเพิ่มสัญลักษณ์

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
	กดปุ่มบันทึกก่อนวาด สัญลักษณ์	ไม่สามารถไปหน้าถัดไปได้
	กดปุ่มบันทึกหลังวาด สัญลักษณ์	สามารถเข้าหน้าแสดงรายละเอียด
ค้นหาสัญลักษณ์	กรอกข้อความที่ต้องการ ค้นหา	ค้นหาไม่ได้

5.3 การแปลงข้อความเป็นเสียงสังเคราะห์ มีการทดสอบดังนี้

ตารางที่ การทดสอบแปลงเสียงภายนอกเป็นข้อความ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
การแปลงเสียงเสียง ภายนอกเป็นข้อความ	วาดสัญลักษณ์บนหน้าจอ	สัญลักษณ์ไม่สามารถดึงรายการข้อความ หรือประโยคมาให้เลือกได้
	กดปุ่มยืนยัน	แสดงหน้าเลือกรายการข้อความหรือ ประโยค
	เลือกข้อความหรือประโยค จากรายการ	แสดงเสียงสังเคราะห์ตามข้อความหรือ ประโยคที่เลือก แต่ส่งเสียงซ้ำ
	กดปุ่มโต้ตอบ	เข้าแสดงหน้ารับเสียง
	กดปุ่มอีกครั้ง	แสดงเสียงสังเคราะห์อีกครั้ง

บทที่ 6

สรุปและข้อเสนอแนะ

จากขั้นตอนการศึกษาระบบ ตั้งแต่เริ่มต้นโดยผ่านกระบวนการต่าง ๆ ได้แก่ การวางแผน การวิเคราะห์และออกแบบระบบ การพัฒนาโปรแกรมและทดสอบโปรแกรมจนกระทั่งสิ้นสุดกระบวนการ สามารถสรุปผลของโครงการได้ดังนี้

6.1 สรุปความสามารถของระบบ

- 1) สามารถแปลงเสียงภายนอกเป็นข้อความ
- 2) สามารถบันทึกสัญลักษณ์แทนข้อความหรือประโยค
- 3) สามารถแปลงข้อความเป็นเสียงสังเคราะห์

6.2 ปัญหาและอุปสรรคในการพัฒนา

จากปัญหาการเก็บ Gesture เพื่อเรียกใช้งานโดย Gesture Package ของ Android ที่มีอยู่แล้ว ดังนั้นสัญลักษณ์จึงต้องเก็บเป็นไฟล์ในหน่วยความจำบนอุปกรณ์ Smart Phone ด้วย GestureLibrary ผู้พัฒนาจึงแยกเก็บข้อมูลของสัญลักษณ์และข้อมูลอื่น ได้แก่ ชื่อสัญลักษณ์ รายละเอียดสัญลักษณ์ และรายการข้อความ แยกกันโดยเก็บข้อมูลอื่นไว้ที่ฐานข้อมูล SQLite

6.3 แนวทางในการพัฒนาต่อ

- 1) พัฒนาให้แอปพลิเคชันมีเมนูแบบ Floating menu บนหน้าจอให้สามารถเลือกเมนูเรียกใช้ได้แม้กำลังใช้งานแอปพลิเคชันอื่นอยู่
- 2) พัฒนาให้แอปพลิเคชันสามารถแสดงผลตามภาษาที่เลือกได้
- 3) พัฒนาให้หน้าผู้ใช้งานให้น่าใช้และมีความใช้ง่ายมากขึ้น

บรรณานุกรม

บทที่ 1 รู้จักกับแอนดรอยด์.สืบค้นเมื่อ 9 มิถุนายน 2560,
จาก <http://www.sourcecode.in.th/articles.php?id=71>
ประเภทของระบบปฏิบัติการแอนดรอยด์.สืบค้นเมื่อวันที่ 9 มิถุนายน 2560, จาก <http://android-basicapp.blogspot.com/2012/12/blog-post.html>
Tutorial :: Android :: ความรู้เบื้องต้น :: Application Component ใน Android
Application.สืบค้นเมื่อ 9 มิถุนายน 2560, จาก <http://tutorial.function.in.th/android/application-component>
[Android Code] การทำ Text To Speech เพื่ออ่านข้อความที่ต้องการ.สืบค้นเมื่อวันที่ 15 มิถุนายน 2560, จาก
<http://www.akexorcist.com/2013/08/android-code-text-to-speech.html>
[Android Code] เปลี่ยนเสียงพูดให้กลายเป็นข้อความด้วย Voice Recognition.สืบค้นเมื่อวันที่ 16 มิถุนายน 2560, จาก <http://www.akexorcist.com/2013/0e-sqlite.html>
Gesture คืออะไร ?.สืบค้นเมื่อวันที่ 16 มิถุนายน 2560, จาก <http://www.thaidroid-appvisor.com/2012/06/what-is-gesture.html>
พื้นฐาน Android Gestures Swipe/flip Sliding ตรวจสอบการ Sliding บนหน้าจอ.สืบค้นเมื่อวันที่ 16 มิถุนายน 2560, จาก <http://www.thaicreate.com/mobile/android-gestures-sliding-screen.html>
GestureOverlayView - Android Widgets Example .สืบค้นเมื่อวันที่ 16 มิถุนายน 2560, จาก <http://www.thaicreate.com/mobile/android-gestureoverlayview.html> 8/android-code-string-speech-recognizer.html
[Android Code] ฐานข้อมูล SQLite ถ้าเข้าใจก็ไม่ใช่เรื่องยาก.สืบค้นเมื่อวันที่ 16 มิถุนายน 2560, จาก <http://www.akexorcist.com/2013/01/android-code-sqlite.html>
Gesture คืออะไร ?.สืบค้นเมื่อวันที่ 16 มิถุนายน 2560, จาก <http://www.thaidroid-appvisor.com/2012/06/what-is-gesture.html>

พื้นฐาน Android Gestures Swipe/flip Sliding ตรวจสอบการ Sliding บนหน้าจอ.สืบค้นเมื่อวันที่ 16 มิถุนายน 2560, จาก <http://www.thaicreate.com/mobile/android-gestures-sliding-screen.html>

GestureOverlayView - Android Widgets Example .สืบค้นเมื่อวันที่ 16 มิถุนายน 2560, จาก <http://www.thaicreate.com/mobile/android-gestureoverlayview.html>

ภาคผนวก

ภาคผนวก ก

คู่มือการติดตั้งแอปพลิเคชันพุดสื่อใจ

ภาคผนวก ข

คู่มือการใช้แอปพลิเคชัน

ประวัติผู้เขียน

ประวัติส่วนตัว

ชื่อ-สกุล	นางสาวนัฐทริกา ชีพันธ์
รหัสประจำตัว	5611402933
วันเกิด	9 พฤศจิกายน 2537
ที่อยู่ที่สามารถติดต่อได้	633 หมู่ที่ 5 ตำบลน้ำอ้อม อำเภอกันทรลักษ์ จังหวัดศรีสะเกษ รหัสไปรษณีย์ 33110
เบอร์โทรศัพท์	095-785-9169
อีเมล	natthrika.ch.56@ubu.ac.th

ประวัติการศึกษา

ระดับประถมศึกษา	โรงเรียนอนุบาลตำราขานนุสาร
ระดับมัธยมศึกษาตอนต้น	โรงเรียนกันทรลักษ์วิทยา
ระดับมัธยมศึกษาตอนปลาย	โรงเรียนกันทรลักษ์วิทยา
ระดับอุดมศึกษา	มหาวิทยาลัยอุบลราชธานี

สารบัญ

กิตติกรรมประกาศ.....	3
บทที่ 1 บทนำ.....	5
1.1 ที่มาและความสำคัญ.....	5
1.2 วัตถุประสงค์.....	5
1.3 ขอบเขตของโครงการ.....	5
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	6
1.5 เครื่องมือที่ใช้ในการพัฒนา.....	6
1.6 ขั้นตอนการดำเนินการ.....	7
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	8
2.1 Android System.....	8
2.2 ความรู้พื้นฐานเกี่ยวกับ Android API Packages หลักที่เกี่ยวข้อง.....	13
2.3 ตัวอย่างโปรเจค Gestures Builder.....	19
บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	20
3.1 ภาพรวมของระบบ.....	20
3.2 Use Case Diagram.....	22
3.3 Class Diagram.....	25
3.4 Sequence Diagram.....	29
3.5 State Diagram.....	33
3.6 Entity Relationship Diagram.....	34
3.7 User Interface Design.....	36
บทที่ 4 การพัฒนาระบบงาน.....	41
4.1 การแปลงเสียงเป็นข้อความ.....	41
4.2 การแปลงข้อความเป็นเสียง.....	44
บทที่ 5 การทดสอบระบบ.....	49
5.1 การแปลงเสียงภายนอกเป็นข้อความ.....	49
5.2 การบันทึกสัญลักษณ์แทนข้อความหรือประโยค.....	49
5.3 การแปลงข้อความเป็นเสียงสังเคราะห์.....	49
บทที่ 6 สรุปและข้อเสนอแนะ.....	50
6.1 สรุปความสามารถของระบบ.....	50
6.2 ปัญหาและอุปสรรคในการพัฒนา.....	50
6.3 แนวทางในการพัฒนาต่อ.....	50
บรรณานุกรม.....	50
ภาคผนวก ก.....	53

คู่มือการติดตั้งแอปพลิเคชันพุดสื่อใจ.....	53
ภาคผนวก ข.....	54
คู่มือการใช้แอปพลิเคชัน.....	54
ประวัติผู้เขียน.....	55

สารบัญตาราง

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

ตารางที่ 3.1 สัญลักษณ์ของ Use Case Diagram

ตารางที่ 3.1 สัญลักษณ์ของ Use Case Diagram (ต่อ)

ตารางที่ 3.2 สัญลักษณ์ของ Class Diagram

ตารางที่ 3.2 สัญลักษณ์ของ Class Diagram (ต่อ)

ตารางที่ 3.3 อธิบายแผนภาพ Class Diagram ของภาพที่ 3.5

ตารางที่ 3.4 สัญลักษณ์ของ Sequence Diagram

ตารางที่ 3.5 สัญลักษณ์ของ State Diagram

ตารางที่ 3.6 สัญลักษณ์ของ ER Diagram

ตารางที่ 3.7 ตารางสัญลักษณ์

ตารางที่ 3.8 ตารางข้อความหรือประโยค

สารบัญรูปภาพ

- ภาพที่ 2.1 โครงสร้างสถาปัตยกรรม Android
- ภาพที่ 2.2 การเรียกใช้งาน Speech Recognition
- ภาพที่ 2.3 การแปลงข้อความจากเสียง
- ภาพที่ 2.4 การกำหนดภาษาสำหรับการรับค่า
- ภาพที่ 2.5 การเรียกใช้งาน Text to Speech
- ภาพที่ 2.6 การตรวจสอบการ Initialize
- ภาพที่ 2.7 คำสั่งสร้างตารางบน Android
- ภาพที่ 2.8 คำสั่งสร้างตารางบน Android
- ภาพที่ 2.9 หน้าหลัก Gestures Builder
- ภาพที่ 2.10 หน้าบันทึก Gestures
- ภาพที่ 2.11 แสดง Dialog เปลี่ยนชื่อหรือลบ Gestures
- ภาพที่ 3.1 ภาพรวมการทำงานของ การแปลงข้อความ เป็นเสียงสังเคราะห์
- ภาพที่ 3.2 ภาพรวมการทำงานของ การแปลงข้อความ ภายนอกเป็นข้อความ
- ภาพที่ 3.3 ภาพรวมการทำงานของ การจัดการสัญลักษณ์ในแอปพลิเคชันพูดสื่อใจ
- ภาพที่ 3.4 Use Case Diagram แสดงการทำงานของแอปพลิเคชันพูดสื่อใจ
- ภาพที่ 3.5 ภาพ Class Diagram ของแอปพลิเคชันพูดสื่อใจ
- ภาพที่ 3.6 Sequence Diagram การแปลงเสียงภายนอกเป็นข้อความ
- ภาพที่ 3.7 Sequence Diagram การเพิ่มสัญลักษณ์
- ภาพที่ 3.8 ER Diagram ของแอปพลิเคชันพูดสื่อใจ
- ภาพที่ 3.9 ภาพหน้าแรกของแอปพลิเคชันพูดสื่อใจ
- ภาพที่ 3.10 ภาพหน้าแรกของข้อความ
- ภาพที่ 3.11 ภาพแสดงรายการข้อความหรือประโยค
- ภาพที่ 3.12 ภาพหน้าแสดงข้อความหรือประโยค
- ภาพที่ 3.13 ภาพส่วนแสดงข้อความจากเสียง
- ภาพที่ 3.14 ภาพแสดงหน้าแรกส่วนบันทึกแสดงรายการสัญลักษณ์
- ภาพที่ 3.15 หน้ากรอกข้อมูลเกี่ยวกับสัญลักษณ์
- ภาพที่ 3.16 หน้าวาดสัญลักษณ์

สารบัญตาราง

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

ตารางที่ 3.1 สัญลักษณ์ของ Use Case Diagram

ตารางที่ 3.1 สัญลักษณ์ของ Use Case Diagram (ต่อ)

ตารางที่ 3.2 สัญลักษณ์ของ Class Diagram

สารบัญรูปภาพ

ภาพที่ 2.1 โครงสร้างสถาปัตยกรรม Android

ภาพที่ 2.2 การเรียกใช้งาน Speech Recognition

ภาพที่ 2.3 การแปลงข้อความจากเสียง

ภาพที่ 2.4 การกำหนดภาษาสำหรับการรับค่า

ภาพที่ 2.5 การเรียกใช้งาน Text to Speech

ภาพที่ 2.6 การตรวจสอบการ Initialize

ภาพที่ 3.1 ภาพรวมการทำงานของ การแปลงข้อความ เป็นเสียงสังเคราะห์

ภาพที่ 3.2 ภาพรวมการทำงานของ การแปลงข้อความภายนอกเป็นข้อความ

ภาพที่ 3.3 ภาพรวมการทำงานของ การจัดการสัญลักษณ์ในแอปพลิเคชันพูดสื่อใจ

ภาพที่ 3.4 Use Case Diagram แสดงการทำงานของแอปพลิเคชันพูดสื่อใจ

ภาพที่ 3.5 ภาพ Class Diagram ของแอปพลิเคชันพูดสื่อใจ

